Requirements

Group 21
Generic Games

Josh Thomas
Andrew Palombo
Oscar Gunn
Scarlet Desorgher
Immanuel Ghaly
Madeleine Nielsen

Extended by

Group 18
Team B

Olivia Betts
Zac Bhumgara
Nursyarmila Ahmad Shukri
Cameron Duncan-Johal
Muaz Waqas
Oliver Northwood
Teddy Seddon

Introduction

Our requirements were elicited through an interview with the customer, and an initial product brief that was presented to us on accepting the project.

Interview

We elicited the requirements through interviews with the customer, where the customer would be asked a series of questions such that the team could gather appropriate details for the requirements of the game, and any core qualities and moralistic details that the game should or shouldn't display. It also allowed us to understand the customer's requirements for the minimum system hardware. By asking a mixture of predetermined close-ended, leading and probing questions, the information gathered was concise, relevant and easy to analyse. Due to the restricted nature of initial answers. However we then used open-ended follow up questions, which provided us with more detailed responses which helped us to better understand the customers perspective. At the end of each question we asked the customer to rate the importance of each requirement that we identified. We did this so that we could allocate our time appropriately and it also helped us in handling conflicting requirements.

Brainstorming

As a group we analysed the information gathered such as the overall context and the target demographic. We then used this information in conjunction with the product brief to produce the user requirements [fig.1]. By identifying the target demographic we ensured that the needs of the users were being satisfied.We then developed the system requirements by going through each user requirement and brainstorming ideas about what the system needs to do, in order for that requirement to be met. This method was used so that the system requirements were traceable, which ensured that we met the needs of the stakeholders and prevented any gaps or inconsistencies.

After the interview, we broke the general points made by the customer in combination with the product brief, to establish different requirements and brainstorm new ones. We then split the requirements into two groups, user requirements (UR) and system requirements (SR).

- User requirements were classified as tasks that users should be able to accomplish using the system.
- User requirements were to be written in plain language without technical components.
- System requirements were classified as the capabilities of the system and how the system will complete tasks established by the user requirements.
- System requirements were intended as a guideline for implementation, this resulted in the preference of technical language over plain language when establishing these.

Conflict Negotiation

During the process of creating system requirements [imozwastaken.github.io/requirements.html#functional , fig.2] we identified some conflicts, caused by the differing needs of the stakeholders. In order to resolve these conflicts we re-evaluated each requirement by considering the project constraints such as time,

technology and personnel and we also consulted the risks involved with each requirement. We then analysed the importance of each requirement to the respective stakeholder and then prioritised the requirements accordingly. Through group discussion we then identified the lower priority requirement and suggested ways to alter it so we could accommodate both requirements. However, if that wasn't possible then the lower priority requirement was removed.

Presentation

The requirements were documented in 3 separate tables for the user requirements [imozwastaken.github.io/requirements.html#user , fig.1], the functional requirements [imozwastaken.github.io/requirements.html#functional , fig.2] and non-functional requirements [imozwastaken.github.io/requirements.html#nonfunctional, fig.3]. The user requirements were also organised by importance, which allowed us to appropriately allocate our time during the implementation. The functional requirements were organised by the user requirement they were traced from and we included an index that references the corresponding user requirement. This ensured that the implementation met the needs of the stakeholders, as we implemented the system directly from the system requirements. For the non functional requirements we included a fit-criteria so that the requirements are verifiable.

Extending requirements

When we (as Group 18) inherited this requirements document from Group 21, the first thing we did was check through the listed requirements to ensure all requirements for both assessment 1 and 2 were listed. From here, we made a couple of changes, editing the wording to clarify sections (see UR_SWITCHING_COOKS) and adding some new requirements (UR_DIFFERENT_RECIPES, UR_POWERUP, UR_DIFFICULTY, UR_SAVESTATES)

After emailing the customer again, we confirmed that these new requirements were satisfactory.

User Requirements

INDEX	ID	DESCRIPTION	PRIORITY
1	UR_SWITCHING_COOKS	The user should be able to switch between 3 cooks	Shall
2	UR_MOVING_COOK	The user needs to be able to move the cook to the cutting, baking, frying and serving stations and the pantry.	Shall
3	UR_COOK_ACTION	The user needs to be able to interact with the stations their cooks are standing near	Shall
4	UR_STATION_ACTION	While using the station the user needs to be able to complete an action (i.e. flipping a patty). If they fail this, they have to restart the step	Shall
5	UR_COOK_STACK	The users needs to be able to stack ingredients from different stations to a cook - essentially a cooks inventory	Needs
6	UR_CUSTOMER_VIEW	The user needs to be able to see the waiting customers, their orders and the recipes of the orders	Shall
7	UR_TIME_CUSTOMERS	User needs to be able to see how much time has elapsed since the customer has placed their order	Needs
8	UR_REPUTATION	The user starts with 3 reputation points (essentially HP). These need to be clearly displayed and possibly can be increased	Shall
9	UR_EARNINGS	Users need to be able to collect earnings. Current earnings shown clearly to the user.	Shall
10	UR_MAX_SERVE	User needs to be able to see their maximum number of customers they have served in endless mode	Shall
11	UR_SCENARIO_TIME	In scenario mode the user should know how long it took them to complete the scenario	Shall
12	UR_USER_EXPERIENCE	The user should be familiar with the design of the game, and it should be simple and intuitive.	Should
13	UR_DIFFERENT_RECIPES	The game should should have salads, burgers, pizzas, and jacket potatoes	Needs
14	UR_GAMEMODES	The user should be able to choose between Scenario Mode or an Endless Mode	Needs
15	UR_POWERUPS	There should be 5 powerups the user can obtain	Needs
16	UR_DIFFICULTY	The user can select between different difficulty levels	Needs
17	UR_SAVESTATES	The user should be able to save the state of the game at any point and resume a saved game later	Should

18	UR_LICENCE	All assets and technologies should be appropriately licenced	Needs
19	UR_READABLE_CODE	All code should be readable and easily manageable. All of the code should be easy for future developers to pick up and understand.	Needs
20	UR_FAMILY_FRIENDLY_CONTENT	The game will not have any graphic content, including swearing or violence	Shall

fig.1

Functional Requirements

INDEX	ID	DESCRIPTION	USER REQUIREMENT
1.1	FR_DIFFERENT_COOKS	The system should be able to differentiate between different cooks	UR_SWITCHING_COOKS
1.2	FR_CURRENT_COOK	The system should be able to have a current cook that is being controlled by the user. Current cook is highlighted	UR_SWITCHING_COOKS
1.3	FR_SWITCH_COOKS	The system should allow the user to switch between cooks by clicking on their respective sprites	UR_SWITCHING_COOKS
2.1	FR_MOVE_COOK	The system should allow the user to move their selected cook by clicking on the stations	UR_MOVING_COOK
2.2	FR_DESTINATIONS	The system should have a set of different coordinates that the cooks can move to	UR_MOVING_COOK
2.3	FR_MOVING_ROUTES	The system should have a set of routes that cook can take between the different stations	UR_MOVING_COOK
2.4	FR_MOVING_GRAPHICS	The system should have graphics for cooks moving around the kitchen	UR_MOVING_COOK
2.5	FR_COOK_COLLISIONS	The system should ensure that two cooks moving on the same path won't affect their movement	UR_MOVING_COOK
3.1	FR_USE_STATION	After the cook has arrived at the serving station or pantry, a menu opens up. After the cook has arrived at a cooking station, the user must click on the station to use it/drop their item.	UR_COOK_ACTION
3.2	FR_DROP_RESTRICTION	The system should not allow the user to drop an ingredient on an incorrect station.	UR_COOK_ACTION
3.3	FR_TAKE_PREPPED_INGREDIENT	Once the preparation step has been completed the system should allow the user to add the prepared ingredient to their stack by clicking on a button.	UR_COOK_ACTION
3.4	FR_COOK_RESTRICTIONS	The system should prevent the user from being able to control the cook during a preparation step.	UR_COOK_ACTION
3.5	FR_VIEW_PANTRY	When a user clicks on the pantry a window should pop up with the various ingredients; the top row of order tickets(with the recipes) should still be visible, as well as the stack on the right.There should also be a bin icon that removes the top item from the stack	UR_COOK_ACTION
3.6	FR_EXIT_PANTRY	There should be an exit button to leave the pantry window.	UR_COOK_ACTION
3.7	FR_SERVING_STATION	When the user clicks on the serving station a menu should pop up with images of the dishes	UR_COOK_ACTION
3.8	FR_EXIT_SERVING_STATION	The system should provide an exit button to leave the serving station	UR_COOK_ACTION
3.9	FR_SERVE_DISH	If the user has the correct ingredients on their stack (in any	UR_COOK_ACTION

		order) the system should allow them to click on the image of the dish to serve it, it is automatically served to the customer that waited to longest and the order ticket is removed from the top row	
3.10	FR_INCOMPLETE_DISH	If the user tries to serve an incomplete dish the system should alert the user what ingredients they are missing.	UR_COOK_ACTION
3.11	FR_BIN	The system should provide a bin which when clicked, removes the top ingredient from their stack.	UR_COOK_ACTION
4.1	FR_STATION_ACTION	The system should display a button when a cooking action is required, such as flipping a burger. There should be a progress bar above the cook showing the progress of the cook's current action.	UR_STATION_ACTION
4.2	FR_ACTION_TIME_LIMIT	The system should provide a time limit for the user to click the button.	UR_STATION_ACTION
4.3	FR_PREP_FAIL	The system needs to destroy the user's ingredient if they fail to complete the task given to them within the allotted time	UR_STATION_ACTION
4.4	FR_AVOID_RESET	The game should actively store the positions of cooks and customers so that when moving from the pantry screen back to the game screen assets aren't moved.	UR_STATION_ACTION
5.1	FR_COOK_STACK	The system should provide a sidebar with a unique stack for each cook and should display the stack of the selected cook.	UR_COOK_STACK
5.2	FR_ADD_TO_STACK	The system should add an ingredient to the stack when the user clicks on an ingredient in the pantry.	UR_COOK_STACK
5.3	FR_STACK_LIMIT	The system should prevent the user from adding more than 5 ingredients on their stack.	UR_COOK_STACK
6.1	FR_CUSTOMER_SPRITES	The system should provide sprites for each customer	UR_CUSTOMER_VIEW
6.2	FR_ORDER_TICKET	There should be a row at the top with the customer order tickets which appears when a new customer arrives. Each ticket should also include the recipe.	UR_CUSTOMER_VIEW
7.1	FR_ORDER_TIME_LIMIT	The system should provide a progress bar at the bottom of every ticket, showing how much time the user has to serve that customer.	UR_TIME_CUSTOMERS
8.1	FR_REP_DISPLAY	The system should display the user's reputation points graphically	UR_REPUTATION
8.2	FR_REP_LOSS	The system should remove a reputation point if the user doesn't serve a customer in time.	UR_REPUTATION
9.1	FR_SCENARIO_MODE_EARNINGS	For scenario-based mode the earnings are proportional given for the difficulty of the order made by each individual customer. E.g. the pizza item might give more earnings than the salad.	UR_EARNINGS
9.2	FR_ENDLESS_MODE_EARNINGS	For endless mode the earnings are given for each customer the user serves.	UR_EARNINGS

9.3	FR_SPEND_EARNINGS	At start, not all stations are available (eg, can only make salad.) Money earned can be used to buy more.	UR_EARNINGS
10.1	FR_CUSTOMERS_SERVED	The system should provide a counter for the number of customers they serve in endless mode.	UR_MAX_SERVE
11.1	FR_SCENARIO_TIME	The system should display the time the user takes when playing in scenario mode.	UR_SCENARIO_TIME
12.1	FR_GAME_OVER	After the game is over the system should display the earnings for that round, the user's balance, a text field so the user can type their name and buttons that take you to the leaderboard or the main menu.	UR_USER_EXPERIENCE
12.2	FR_LEADERBOARD	The system should display the top 10 scores on the leaderboard along with the user's name.	UR_USER_EXPERIENCE
12.3	FR_MAIN_MENU	The main menu should have buttons for scenario mode, endless mode.	UR_USER_EXPERIENCE
12.5	FR_SFX	The system should provide some sound effects, and background music.	UR_USER_EXPERIENCE
14.1	FR_CONFIGURE_CUSTOMERS	The system should allow the user to select the amounts of customers visiting in scenario mode (with a default of 5)	UR_GAMEMODE
14.2	FR_CUSTOMER_ARRIVALS	In Endless Mode, users will initially arrive one at a time, but may end up arriving in groups of two or three	UR_GAMEMODE
15.1	FR_POWERUP_EFFECT	The powerups should have a noticeable but temporary positive effect on the chef that picked it up	UR_POWERUPS
15.2	FR_POWERUP_APPEARANCE	Powerups should appear at random intervals, and be able to be clicked on to be picked up	UR_POWERUPS
16.1	FR_SELECT_DIFFICULTY	Before starting endless mode, the user should be able to select difficulty level that will affect different elements of gameplay (easy, medium, or hard)	UR_DIFFICULTY
16.2	FR_TIME_DIFFICULTY	Depending on the level of difficulty the amount of time that customers stay at the counter should appropriately change.	UR_DIFFICULTY
16.3	FR_POWERUP_DIFFICULTY	Depending on the level of difficulty the length that power ups are applied should appropriately change.	UR_DIFFICULTY
16.4	FR_LIVES_DIFFICULTY	Depending on the level of difficulty the amount of lives (reputation points) has should appropriately change.	UR_DIFFICULTY
17.1	FR_SAVE_GAME	There should be a pause menu, which allows the user to save the game	UR_SAVESTATES
17.2	FR_LOAD_GAME	The main menu should have a button to allow the user to load a previously saved game	UR_SAVESTATES

fig.2

Non-Functional Requirements

ID	DESCRIPTION	USER REQUIREMENT	FIT CRITERIA
NFR_PORTABILITY	It should be able to run on different operating systems.	UR_USER_EXPERIENCE	It should be able to operate on at least 2 different operating systems
NFR_VISUAL_CUE	When the user needs to perform a station action, the system should provide the user with a visual cue that shouldn't rely on text or colour	UR_STATION_ACTION	80% of testers were able to identify and respond to each visual cue.
NFR_TIMING	The Game should take 5-10 minutes.	UR_USER_EXPERIENCE	5≤ mean time ≤ 10
NFR_USABILITY	The system should be easy to understand, and use.	UR_USER_EXPERIENCE	80% of testers would agree that the system was easy to use.
NFR_PERFORMANCE	The system should be able to run for an entire day, without any major drops in performance.	UR_USER_EXPERIENCE	The system can run for at least 8 hours.
NFR_FAILURE_RATE	The system should not crash or freeze on a standard computer	UR_USER_EXPERIENCE	The system should crash no more than 1% of the time
NFR_LOAD_TIME	The system should load quickly	UR_USER_EXPERIENCE	Maximum time to load should be 20 seconds.
NFR_USER_ENGAGEMENT	The system should be fun and engaging to prospective students	UR_USER_EXPERIENCE	80% of testers would describe the system as fun and engaging
NFR_OPERABILITY	Ensure new users can play the game without a tutorial or a rules page	UR_USER_EXPERIENCE	80% of testers completed the game without assistance
NFR_FRAME_RATE	The frame rate should not drop during expected usage	UR_USER_EXPERIENCE	Minimum frame rate of 20fps
NFR_VISIBILITY	The system should use a colour scheme that maximises visibility	UR_USER_EXPERIENCE	80% of testers found that the colour scheme didn't hinder visibility
NFR_ACTION_TIME	The cooking actions shouldn't take too long so the user doesn't become unengaged	UR_COOK_ACTION	Maximum time for a cooking action is 30 seconds
NFR_SERVING_TIME	The majority of users should be able to serve the dish in time	UR_COOK_ACTION	80% of testers were able to serve the order in time.
NFR_MOVE_TIME	The time it takes for a cook to move to a station should be reasonable.	UR_MOVING_COOK	Maximum time for a cook to reach a station should be 5 seconds

NFR_TEXT_SIZE	Ensure that any text is of the appropriate font size	UR_USER_EXPERIENCE	Minimum font size of 7
NFR_SPRITE_VISIBILITY	Sprite should be appropriately sized and should not overlap	UR_CUSTOMER_VIEW	Minimum size for a sprite should be 5*5
NFR_AVAILABILITY	The system should be readily available	UR_USER_EXPERIENCE	The system should be available 99% of the time.
NFR_WINDOW_RESIZE	The game should appropriately resize with changes in the window size	UR_USER_EXPERIENCE	The entire game screen should be visible within any given size of application window
NFR_CODE_READABILITY	The code should be generally well structured and presentable	UR_READABLE_CODE	There should be appropriate white space and indentation for every line of code
NFR_REFACTORED_LONG_METHODS	Longer methods shall always be refactored into smaller more easily manageable methods	UR_READABLE_CODE	No method should be longer than approx. 1 page
NFR_GRAPHIC_CONTENT	The game shall not contain any graphic swearing or violence as the game should be appropriate for people of all ages.	UR_FAMILY_FRIENDLY_CONTENT	Any given tester shouldn't feel offended by the content of the game, allowing the game to be appropriate for people of all ages.

fig.3

References

1) https://medium.com/omarelgabrys-blog/requirements-engineering-introduction-part-1-6d49001526d3
2) https://link.springer.com/article/10.1007/s10664-011-9156-x
3) https://www.cs.odu.edu/~zeil/cs350/latest/Public/eliciting/index.html#:~:text=Eliciting%20requirements%20is%20the%20process,more%20formal%20requirements%20document%20later.
4)